

# Bab 1

## **Apa Itu Website dan Mengenal Cara Kerjanya**

Website atau situs web, kini telah menjadi sesuatu yang penting di zaman internet ini. Jika zaman dahulu website hanya lazim dimiliki oleh perusahaan-perusahaan besar, kini website sudah lazim dimiliki oleh perorangan bahkan kadang satu orang punya lebih dari satu website.

Karena itu penting sekali bagi Anda untuk mengetahui apa itu website, bagaimana cara kerja website hingga bisa menampilkan halaman yang bisa dilihat di browser dan digunakan untuk banyak keperluan, seperti menampilkan informasi, berbelanja online hingga narsis via social media.

Website punya berbagai variasi. Tapi apa pun jenisnya, website selalu punya satu ciri, yaitu terdiri atas halaman-halaman web/web pages. Mengenai jumlah webpage, tiap website berbeda-beda, tergantung pada kebutuhan. Ada yang bisa dihitung dengan jari, hingga yang ratusan atau ribuan halaman bahkan lebih dari itu.

Saat membuat website, biasanya web master membuat halaman web satu demi satu, baik secara manual ataupun menggunakan software manajemen konten. Di bab pertama ini, Anda akan mengenal apa itu website dan cara kerjanya.

### **1.1 Apa Itu Website**

Website adalah kumpulan halaman web yang diletakkan dalam satu tempat/site/situs. Jadi, di dalam website terdapat halaman-halaman

web, beserta file-file pendukungnya, seperti gambar, video, animasi, dan file-file digital lain yang diletakkan dalam satu tempat yang diidentifikasi melalui nama domain (domain name) dan alamat ip (IP address).

Nantinya website di-host/diletakkan di minimal satu web server. Untuk mengakses webserver, diperlukan koneksi internet, atau koneksi intranet (untuk web server lokal di jaringan). Jadi, komputer pengakses (client) mengakses komputer tempat terletak web server (server) dan kemudian mengakses halaman-halaman web tersebut.

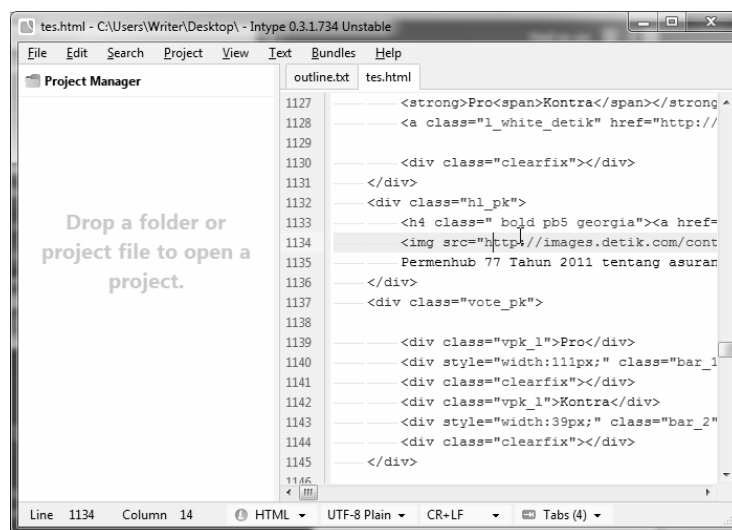
Komputer server ini biasanya speknya lebih tinggi dibandingkan komputer biasa karena komputer server ini biasanya memberikan layanan/service web kepada client. Karena itulah harus punya sumber daya yang lebih besar dibandingkan dengan komputer biasa.



*Gambar 1.1 Contoh komputer server dalam bentuk rak*

Sebuah halaman web lazim disebut dokumen. Dokumen ini bisa dibuat oleh tool apa pun yang mendukung kegiatan scripting. Ini karena dokumen web menggunakan prinsip terbuka. Dokumen halaman web ini ditulis dalam bentuk plain text/teks murni.

Aturan pemformatan dokumen halaman web ini mematuhi aturan HTML Hypertext Markup Language (HTML) yang kemudian disempurnakan dengan paradigma XML (Extensible Markup Language) menjadi XHTML (Extensible Hypertext Markup Language).



**Gambar 1.2** Intype, salah satu software text editor

Untuk mentransfer dokumen halaman web ke client, digunakan teknologi protokol HTTP (*HyperText Transfer Protocol*). Protokol HTTP ini juga dapat ditingkatkan keamanannya menggunakan enkripsi sehingga menjadi HTTPS yang memungkinkan halaman website dipakai untuk aplikasi-aplikasi yang memerlukan keamanan lebih tinggi, seperti e-commerce ataupun online banking.

Untuk mengakses halaman web, komputer user harus memiliki software yang disebut browser. Website-website yang ada di seluruh dunia ini membentuk suatu “dunia maya” yang disebut World Wide Web dan sering disingkat sebagai WWW.

Untuk mengakses sebuah website, diperlukan alamat penentu lokasi website tersebut. Jadi, browser nanti membuka website sesuai dengan alamat yang berlaku. Jika website dengan alamat tersebut ada, maka halaman sesuai dengan url yang bersangkutan akan dibuka. Alamat dari website atau webpage ini disebut sebagai **URL** yang merupakan singkatan dari *Uniform Resource Locator*. Bagian domain utama dari url website disebut **homepage**.



**Gambar 1.3** Tampilan browser ketika membuat website dengan alamat domain name = koprol.com

Untuk menuju ke halaman-halaman webpage lain di sebuah website, homepage dari website tersebut biasanya memberi menu navigasi yang memungkinkan pengunjung lebih mudah menuju ke bagian-bagian tertentu dari website. Menu navigasi ini juga memungkinkan pengunjung mengenal struktur website dengan lebih baik.

Umumnya, setiap pengunjung website bisa mengakses semua bagian dari website tersebut. Kecuali beberapa bagian yang diset untuk administrator atau orang yang terotorisasi. Bagian khusus admin atau yang terotorisasi ini biasanya mensyaratkan username dan password untuk login ke dalamnya. Contoh situs yang memerlukan otorisasi misalnya situs bisnis, situs berita, situs akademis, situs gaming, situs forum, bulletin board, webmail, situs social networking, dan sebagainya.

### 1.1.1 Mengenal Tipe Web Page

Ditinjau dari cara kerjanya, tipe webpage ada dua, yaitu statis dan dinamis. Apa perbedaannya? Webpage statis adalah webpage yang halamannya selalu memiliki tampilan yang sama, saat diakses oleh bermacam-macam user. Di samping itu, web page statis, pemrosesannya dari server juga langsung, tanpa mengalami parsing.

Adapun webpage dinamis bisa menghasilkan tampilan yang berbeda-beda tergantung pada kondisi user atau kondisi server. Tampilan statis memungkinkan informasi yang ditampilkan kepada user selalu sama karena tidak berubah/statis. Secara konteks, isi dokumen selalu sama.

Teknik statis ini lebih kuno, namun webpage statis juga memiliki kelebihan dibandingkan dengan yang dinamis, antara lain:

1. Mudah dan cepat diimplementasikan, bahkan oleh orang yang baru pertama kali membuat website.
2. Tersedia banyak software yang memungkinkan pembuatan website statis secara cepat. Dengan secepat kilat, Anda dapat membuat website dengan mudah.
3. Cocok digunakan untuk mendemonstrasikan bagaimana cara kerja website dan webpage.
4. Cocok disimpan di cache komputer klien, karena isinya akan selalu sama.

Namun, halaman statis juga memiliki beberapa kekurangan. Berikut ini beberapa kekurangan tipe halaman web statis:

1. Menyulitkan maintenance jika ukuran website sangat besar.
2. Sulit untuk membuat webpage terkesan up to date. Karena jika ingin meng-update tampilan setiap hari, maka Anda juga harus membuat halaman web yang baru setiap hari.
3. Kostumisasi ke pengunjung kurang terakomodatif, karena kostumisasi ini memerlukan script dinamis jenis client side. Jadi, tampilan untuk semua user akan sama.

Website dinamis sepertinya memang cenderung lebih banyak digunakan sekarang ini. Namun, pasti ada kelebihan webpage statis yang menyebabkan website ini tetap juga ramai digunakan hingga

sekarang. Keunggulan utama adalah loading kontennya tidak memakan banyak resource sehingga performa loadingnya bagus, cepat dimuati di browser, jauh lebih cepat dibandingkan halaman dinamis. Bahkan beberapa web server seperti Apache dioptimalkan untuk meng-host webpage statis.

Pernah suatu ketika, situs ubuntu.com dibuat menggunakan CMS yang merupakan situs dinamis. Ketika rilis ubuntu baru dikeluarkan, maka situs ubuntu menjadi hang, karena banyaknya akses ke situs tersebut yang berbasis dinamis. Namun semenjak dikembalikan ke webpage statis, maka masalah tersebut langsung tidak dihadapi.

Selain itu, halaman web statis akan lebih bisa diandalkan/reliable. Halaman web statis tidak memerlukan pengaturan yang terlalu banyak, bahkan web server saja sudah bisa menjalankan webpage statis. Beda kalau webpage dinamis, harus ada server, lalu ada software untuk mem-parsing halaman tersebut dan sebagainya. Jika pengaturan salah, bisa-bisa tampilan webpage dinamis menjadi error.

Jadi, webpage statis ada keunggulan dan kekurangannya. Dan webpage statis tidak bisa mengakomodasi semua kebutuhan. Jika informasi yang akan Anda sampaikan di webpage kecenderungannya tidak berubah dalam jangka waktu yang lama, maka gunakan tipe halaman web statis.

Tipe webpage yang kedua adalah tipe dinamis. Webpage dinamis dinamakan dinamis karena bersifat “bisa berubah”. Yaitu tampilan di website berubah tergantung pada perilaku client ataupun server. Konten yang ditampilkan akan berubah merespon konteks dan kondisi tertentu.

Ada dua versi pembuatan webpage dinamis, yaitu:

1. Menggunakan bahasa scripting client side untuk mengatur perilaku dari bagian-bagian tertentu di webpage, yang nantinya akan merespon input ataupun keyboard ataupun mouse yang digerakkan oleh user.

Scripting ini dinamakan client side, karena perilaku user sebagai client di browsernya langsung ditangkap dan direspon tanpa harus diproses di server. Contoh bahasa pemrograman client side adalah javascript dan vbscript/asp script.

2. Menggunakan bahasa scripting server side untuk mengubah source code yang ditampilkan sehingga memungkinkan web-page memiliki tampilan yang berbeda-beda tergantung pada hasil pengolahan kode program aslinya di server yang telah di-parsing oleh software parser di server tersebut. Contoh bahasa pemrograman server side adalah PHP, ASP.NET, Coldfusion, Java server pages dan sebagainya.

Kedua teknik di atas akan bisa menghasilkan tampilan yang dinamis, yaitu yang bisa berubah dan adaptif. Dan kedua teknik tersebut bisa dipakai secara bersamaan. Jadi, tidak harus client saja atau server saja, namun Anda juga bisa membuat halaman web dinamis yang mengakomodasi client dan server.

Untuk memakai scripting client side, halaman web harus menggunakan teknologi yang disebut bahasa pemrograman client-side. Bahasa pemrograman client side ini bisa dipakai untuk membuat antarmuka yang lebih kaya. Contoh bahasa pemrograman client side ini antara lain JavaScript dan ActionScript yang lazim digunakan untuk memberi efek Dynamic HTML (DHTML) di halaman web dan Flash.

Scripting client side sekarang juga bisa digabung dengan server-side untuk menghadirkan halaman yang interaktif, namun memungkinkan pengaksesan data di server yang sering disebut remote scripting atau istilah modern-nya Ajax.

Adapun halaman web dinamis server side biasanya menggunakan bahasa pemrograman server-side seperti PHP, Perl, ASP, ASP.NET, JSP, ColdFusion Ruby on Rails, dan sebagainya.

Bahasa-bahasa ini biasanya menggunakan CGI yang merupakan singkatan dari *Common Gateway Interface* untuk menghasilkan halaman web dinamis. Tipe halaman dinamis yang client side merupakan tipe halaman dinamis, di mana interaktivitas atau kedinamisan ini diakibatkan kode yang diakses oleh browser dan disimpan di komputer klien.

Jadi, browser yang digunakan untuk membuka halaman mengambil halaman dari server, lalu memproses kode yang terdapat di halaman tersebut. Dan mengeksekusi kode tersebut sehingga ada perubahan tampilan sesuai yang diinginkan di server.

## 1.2 Cara Kerja dan Jenis Website

Agar bisa memahami cara kerja web, Anda harus mengetahui istilah-istilah dunia web.

1. HTML - HTML singkatan dari HyperText Markup Language. "markup language" adalah bahasa komputer yang menentukan bagaimana page harus diformat. Jika Anda ingin menampilkan teks dengan tulisan biasa, maka Anda tidak perlu HTML, tapi jika Anda ingin mengatur teks dengan font tertentu, warna tertentu, membuat headline, dan memasang gambar, maka Anda perlu menggunakan HTML yang merupakan bahasa untuk mengatur ini semua.
2. Web browser - Web browser seperti Google Chrome, Mozilla Firefox atau Microsoft Internet Explorer, adalah program komputer yang melakukan dua hal. Pertama adalah pergi ke webserver di internet kemudian meminta (*request*) halaman web, sehingga halaman web ini bisa ada di komputer Anda. Kedua, web browser menginterpretasikan tag HTML di halaman untuk menampilkan halaman web sebagaimana yang dikehendaki oleh pembuat atau desainer halaman web itu.
3. Web server - Web server adalah software komputer yang merespon permintaan webpage dari browser dan kemudian memberikan apa yang diminta oleh browser tersebut melalui internet. Web server ini mirip dengan rumah susun, di mana di tiap rumah meng-host webpage-webpage tertentu. Agar webpage bisa di-host di sana, maka harus membayar biaya bulanan. Jadi, webserver ini adalah tempat kost atau kontrakan dari webpage. Pemilik kontrakan inilah yang disebut host dan istilah perkontrakan ini yang disebut hosting.

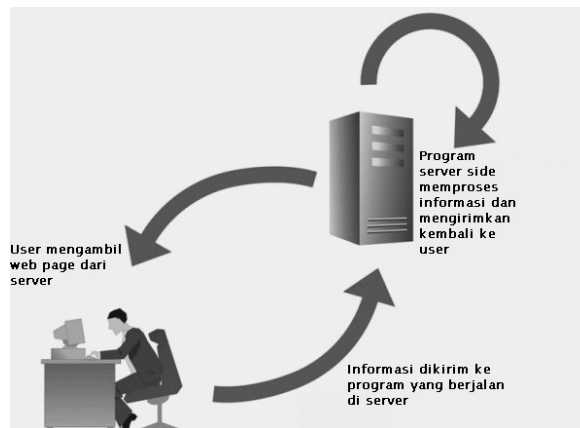
Tapi kalau Anda membuat halaman web yang statis, Anda bisa mencoba-coba membuat halaman web tanpa harus menggunakan web server, alias di file komputer Anda sendiri.

### 1.2.1 Cara Kerja Client - Server

Adapun konten server-side lebih sedikit kompleks cara kerjanya. Terdiri atas beberapa tahapan berikut:



1. Klien mengirimkan permintaan ke server.
2. Server menerima permintaan tersebut kemudian memproses script server side sesuai bahasa pemrograman yang diinginkan.



**Gambar 1.4 Cara kerja bahasa pemrograman server side dalam memproses web berbasis server**

Variasi ketiga adalah perpaduan antara scripting server dan scripting client. Variasi teknik ini memiliki implementasi terbaru yang disebut Ajax yang merupakan singkatan dari *asynchronous javascript and xml*. Ajax bisa dipakai untuk mengembangkan konten yang bisa di-update secara dinamis menggunakan kode client side, namun memanfaatkan data yang ada di server.

Dengan Ajax, webpage bisa meng-update konten di dalamnya tanpa melakukan reload atau refresh halaman web. Contoh aplikasi yang menggunakan ajax ini antara lain berbagai layanan google, seperti Google Maps, Google Docs, dan sebagainya. Di samping itu, hampir semua raksasa IT menggunakan layanan ini, seperti Yahoo, Facebook dan Microsoft.

Berbagai bahasa pemrograman web seperti JSP, PHP, Perl, ASP, Yuma dan Cold Fusion juga memiliki kemampuan mengembangkan web dinamis. Metode dinamis lain adalah metode sinkronisasi melalui sindikasi. Dengan sindikasi melalui RSS (*rich side syndication*) atau format lain berbasis XML, sebuah website dapat men-sindikasi isi

website lain secara dinamis. Jadi, jika konten sumber feed tersebut di-update, sindikasi juga langsung ter-update.

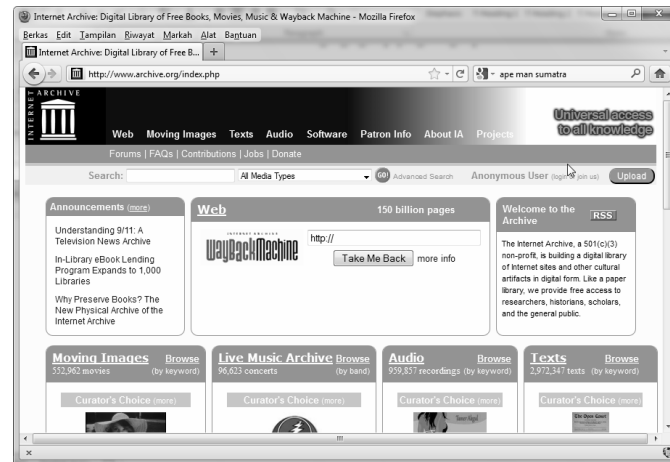
Jika menggunakan teknologi tambahan seperti flash, maka untuk menghadirkan aspek “dinamisme” tersebut, perlu ditambahkan software add-ons di browser. Misalnya di Firefox memerlukan add-ons flash viewer yang memungkinkan konten-konten interaktif dan dinamis dari flash dilihat dengan benar dari firefox.

Selain itu ada lagi teknologi lain, seperti Microsoft Silverlight, Adobe Flash, Adobe Shockwave, Java, dan sebagainya. Contoh aspek dinamis lagi adalah dalam pembuatan iklan untuk website, seperti kode iklan dari google AdSense atau dari software periklanan seperti Openx/PHPAdsNew.

### **1.2.2 Jenis Website**

Semakin berkembangnya internet, juga membuat jenis website yang muncul makin beragam. Jika dahulu website cuma sekadar brosur pasif menampilkan informasi teks dengan sedikit gambar, kini website bahkan bisa dijadikan tempat gaul dan sosialisasi. Apa saja jenis website? Berikut ini beberapa jenis website di zaman sekarang ini.

1. Website Affiliate: Portal yang tidak sekadar berupa CMS, namun juga bisa men-sindikasi konten dari pihak ketiga. Contohnya Yahoo! dan Ebay.
2. Website arsip, website ini menyimpan informasi-informasi penting di internet. Jadi, fungsinya mirip dengan perpustakaan online yang menyimpan hal-hal penting dan menarik di internet. Contohnya Internet Archive (archive.org) yang menyimpan triliunan halaman web, serta google groups yang mengumpulkan data-data lama dari forum usenet.



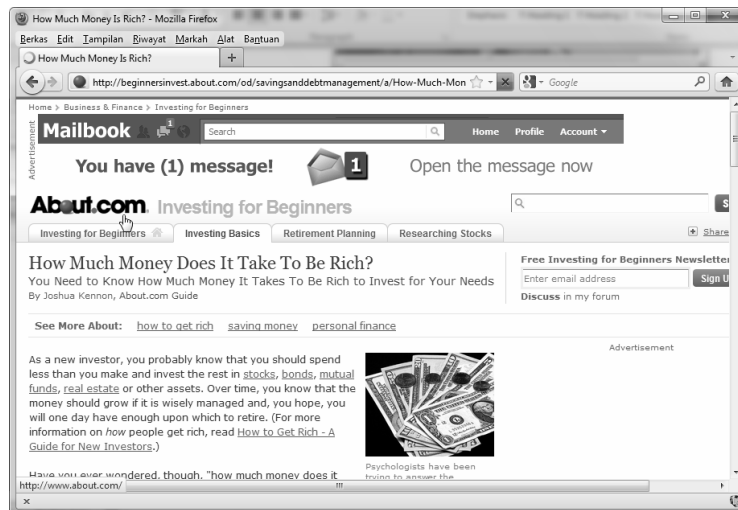
**Gambar 1.5** Archive.org, website arsip yang menyimpan banyak informasi di internet

3. Website blog: Digunakan untuk menyimpan informasi yang dimiliki oleh seorang individu atau perusahaan dan memungkinkan informasi tersebut diberi komentar oleh pengunjung. Contoh penyedia website blog ini misalnya blogdetik.com dan kompasiana.com.



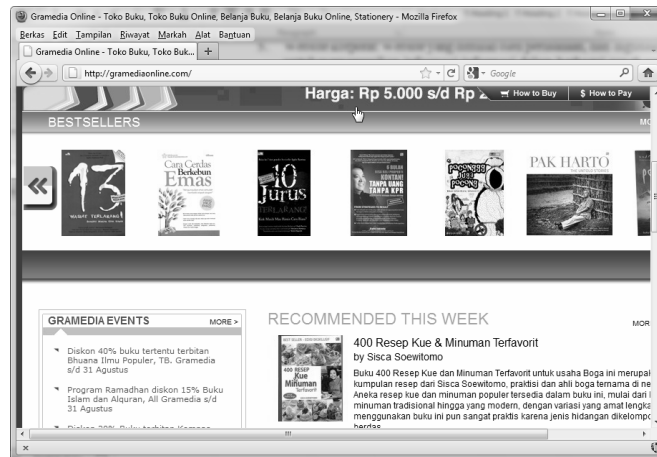
**Gambar 1.6** Kompasiana.com, salah satu penyedia blog yang cukup populer serta simpel dan mudah pengoperasiannya

4. Website konten: Website yang menawarkan informasi yang berguna ke pembacanya. Misalnya Slate.com atau About.com.



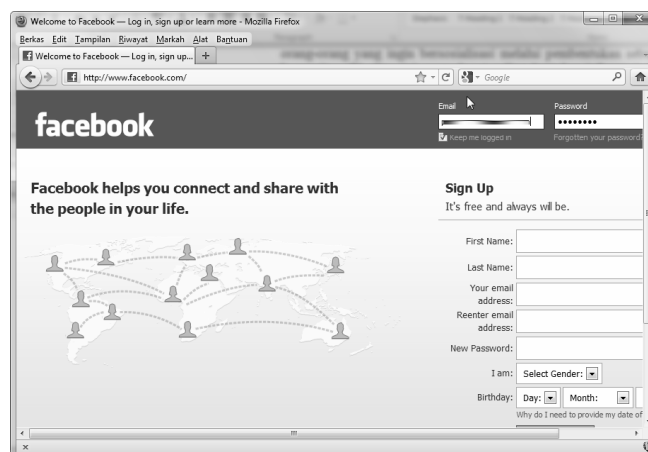
**Gambar 1.7** *About.com, salah satu contoh website konten yang berisi banyak informasi menarik yang ditulis oleh para ahli*

5. Website korporat: Website yang dimiliki oleh perusahaan, dan digunakan untuk menyampaikan informasi-informasi dalam berbagai aspek yang dimiliki oleh perusahaan atau organisasi tersebut.
6. Website e-commerce: Sering juga disebut website toko online yang memungkinkan terjadinya transaksi barang dan jasa melalui website tersebut.



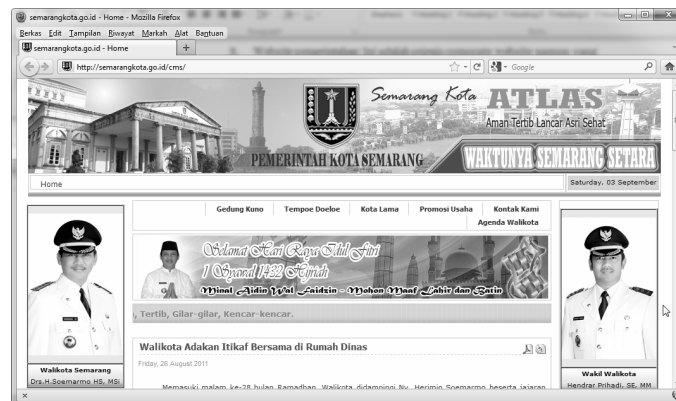
**Gambar 1.8** Contoh toko online yang dibangun menggunakan oscommerce gramediaonline.com

7. Situs komunitas dan social media: Tempat mangkal dan berkumpulnya orang-orang yang ingin bersosialisasi melalui pembentukan sebuah komunitas. Di situs ini, user dapat saling bertukar informasi dan membentuk group-group untuk bersosialisasi. Contohnya yang paling beken adalah Facebook.



**Gambar 1.9** Facebook, situs komunitas paling beken sekarang ini

8. Website pemerintahan: Ini adalah sejenis corporate website namun yang dimiliki oleh institusi pemerintahan. Isinya mengenai dunia birokrasi dan kemajuan-kemajuan yang sudah dilakukan oleh institusi tersebut. Misalnya <http://semarang.go.id> atau <http://polri.go.id>.



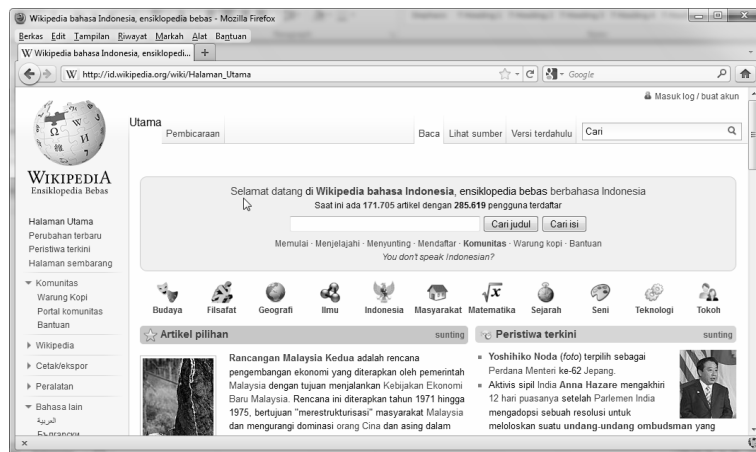
Gambar 1.10 Homepage dari situs resmi kota Semarang

9. Forum: Tempat anggota-anggota yang terdaftar di forum untuk mendiskusikan berbagai topik yang memungkinkan user saling berbagi informasi. Misalnya <http://kaskus.com>, <http://forum.detik.com> dan sebagainya.



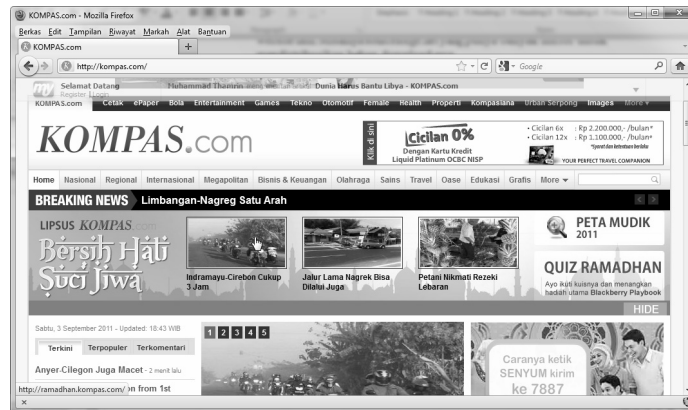
Gambar 1.11 Website forum dari Kaskus.us

10. Website humor: Menampilkan cerita-cerita atau gambar lucu yang bertujuan untuk menghibur.
11. Website informasi: Menampilkan informasi-informasi kepada user, ada yang bersifat komersil dan ada juga yang nirlaba. Misalnya <http://kamus.itb.ac.id> atau <http://wikipedia.org>.



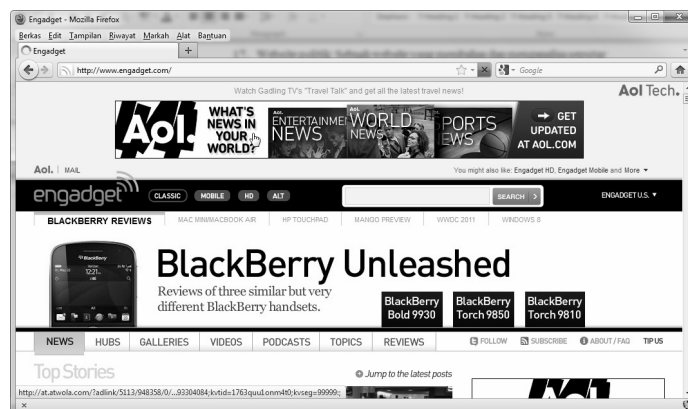
**Gambar 1.12 Website wikipedia, ensiklopedia online**

12. Situs aplikasi: Situs yang digunakan untuk menjalankan sebuah aplikasi, misalnya google docs atau situs-situs yang memakai applet java.
13. Website mirror: Yaitu website yang menampilkan salinan komplet dari website lain. Misalnya sourceforge.net yang punya banyak mirror untuk mendistribusikan beban download-nya.
14. Website berita: Menampilkan berita-berita terkini, seperti <http://detiknews.com>, <http://tvone.co.id>, <http://kompas.com> dan lain sebagainya. Beberapa situs berita juga memungkinkan pengakomodasian komentar oleh pengunjung.
15. Website pribadi: Website yang dimiliki oleh perorangan. Ini bisa berisi tentang semua hal berkaitan dengan kehidupan seseorang. Blog kadang juga bisa digolongkan sebagai bagian dari website pribadi.



**Gambar 1.13 Kompas.com, salah satu website news di Indonesia**

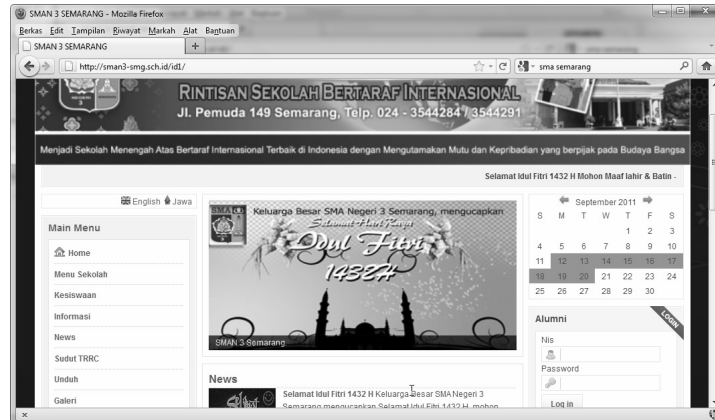
16. Website phishing: Ini adalah website tipu-tipu yang digunakan untuk menipu orang lain. Misalnya mencuri informasi password, dan nomor kartu kredit dengan membuat website yang tampilannya mirip dengan tampilan asli namun berbeda. Contohnya dulu ada website yang mencontoh klikbca.com.
17. Website politik: Sebuah website yang membahas dan menganalisa seputar politik. Contohnya politikana.com.
18. Website review: Website di mana ditampilkan produk-produk beserta penjelasannya, dan pengunjung atau user yang sudah terdaftar bisa me-review produk atau layanan tersebut.



**Gambar 1.14 Situs Engadget.com yang berisi review tentang gadget**

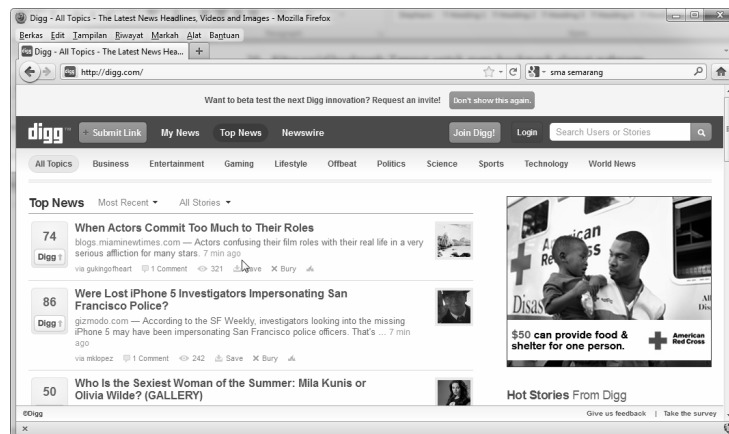


19. Website sekolah: Situs milik sekolah, yang mengakomodasi informasi-informasi seputar sekolah di Indonesia. Ekstensi situs ini adalah .sch.id.



**Gambar 1.15 Fasilitas e-learning dari situs sebuah sekolah**

20. Situs social bookmark: Tempat untuk mem-bookmark alamat webpage tertentu dan biasanya dilengkapi dengan fasilitas rating dan komentar.



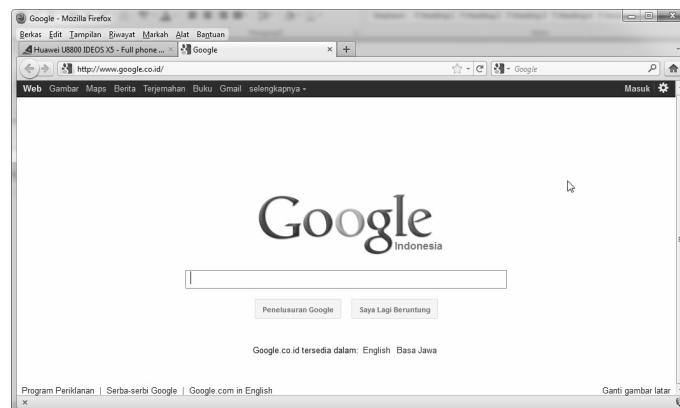
**Gambar 1.16 Website bookmarking dengan digg**

21. Website video sharing: Di mana user dapat meng-upload video ke website dan kemudian website melakukan pemrosesan video dan mengonversinya ke format flash video (.flv) untuk kemudian pengunjung dapat melihat video tersebut dengan gratis. Contoh website tipe ini adalah YouTube dan Google Video.



*Gambar 1.17 Youtube.com untuk video sharing*

22. Website search engine: Digunakan untuk mencari informasi di internet. Website search engine ini memungkinkan pengunjung mencari informasi dengan memasukkan kata kunci pencarian. Biasanya search engine mendukung pencarian teks, gambar, video dan sebagainya. Contohnya Google dan Yahoo!



*Gambar 1.18 Google, salah satu mesin pencari populer di Indonesia*

Selain website-website di atas, masih ada jenis website lainnya. Dan ini akan terus berkembang seiring berkembangnya teknologi pemrograman website dan kreativitas manusia pengguna internet serta para developer pengembang program.

## 1.3 Kode Sumber Halaman dan Tag HTML

Tag HTML adalah isi source code Anda. Ini merupakan elemen kode yang memberitahukan browser apa yang harus dilakukan dengan teks Anda.

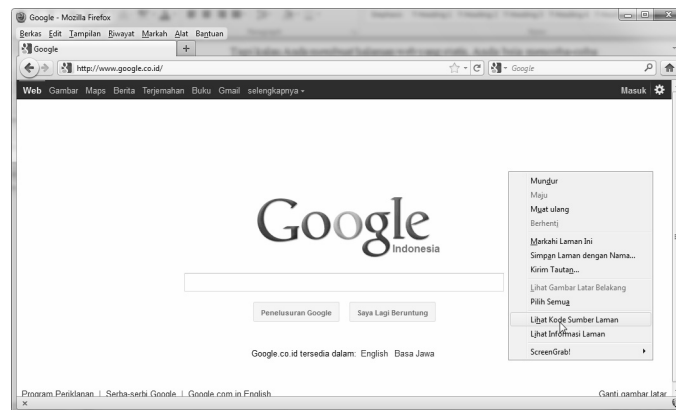
### 1.3.1 Melihat Source Halaman

Bagian inti sebuah web page adalah source code atau kode sumber dari sebuah halaman. Karena kode sumber inilah yang menampung kode HTML dan menentukan bagaimana sebuah halaman ditampilkan.

Kode HTML ini merupakan teks murni yang diketikkan oleh pengembang web dan diinterpretasikan oleh browser untuk memproduksi web page yang diharapkan oleh pengembangnya.

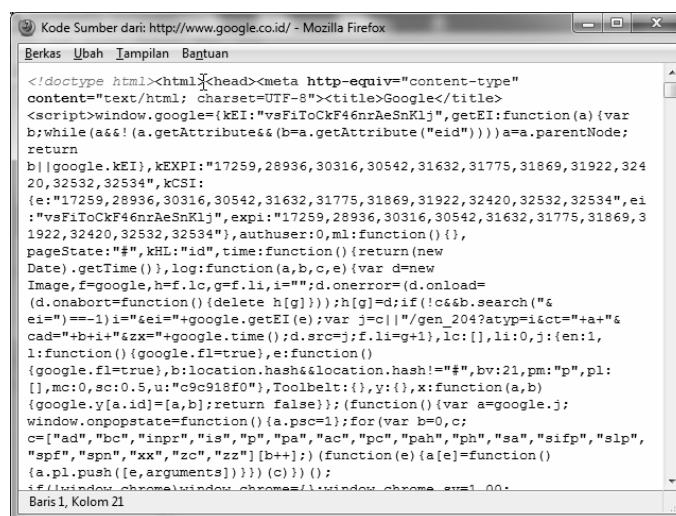
Cara untuk melihat source halaman sangat mudah. Caranya hampir sama di semua browser. Untuk browser firefox misalnya seperti berikut.

1. Buka halaman di browser.
2. Kalau sudah terbuka, klik kanan pada halaman kemudian klik menu **Lihat kode sumber halaman**. Atau dalam bahasa Inggris **View source code**.



**Gambar 1.19** Menu klik kanan untuk membuka kode sumber halaman

3. Maka terlihat source code yang merupakan kode HTML beserta script-script javascript untuk menampilkan tampilan ini.



**Gambar 1.20** Tampilan source code HTML dan javascript

Dengan menggunakan fasilitas view source code, Anda bisa melihat di belakang layar tiap halaman web. Karena itulah halaman web kode sumber di sisi client selalu terbuka. Sementara source code bahasa pemrograman sisi server tetap tidak terlihat.

Dengan fasilitas melihat source code ini, Anda bisa berlatih menjadi desainer web, karena Anda bisa tahu *begini toh* source code dari website-website lain di internet.

### 1.3.2 Tag HTML

Tiap teks muncul di tengah tanda lebih kecil dan lebih besar <teksnya >. Contohnya seperti berikut:

```
<center>  
<body>
```

Untuk memberitahukan browser bahwa tag ini sudah berakhir, Anda harus menggunakan tag penutup, tag penutup sama seperti tag pembuka, hanya saja ada tanda /. Contohnya:

```
</center>  
</body>
```

Tapi ada beberapa tag yang tidak punya tag penutup yang terpisah, misalnya:

```
<br/>
```

Berikut ini beberapa tag standar di halaman web:

1. <HTML>: memberitahukan web browser bahwa ini adalah awal dari dokumen HTML.
2. <HEAD>: memberitahukan web browser bahwa area di sini merupakan header. Di dalam header nanti biasanya ada judul.
3. <TITLE>: memberitahukan web browser bahwa ini adalah judul halaman.
4. <BODY>: memberitahukan web browser bahwa ini adalah awal dari konten halaman. Semua yang terlihat di browser terlihat di sini.
5. Untuk mengakhiri halaman, tag yang diperlukan adalah:

```
</BODY>  
</HTML>
```

## 1.4 Sekilas HTML

Dari sekian banyak tag yang ada. Tag-tag yang wajib ada di setiap webpage adalah tag yang mendefinisikan struktur halaman. Tag ini wajib hukumnya ada di setiap halaman web. Untuk mendefinisikan struktur dokumen HTML, bagian awal dari semua kode harus diawali dengan kode `<html>` dan kemudian ditutup dengan kode `</html>`.

### 1.4.1 Mendefinisikan Struktur Halaman

Jadi, semua kode di antara tag inilah yang dianggap sebagai dokumen HTML. Formatnya seperti berikut.

```
<html>
Semua kode di sini, akan dianggap dokumen HTML
</html>
```

Di dalam tag `<html>` tersebut yang merupakan dokumen HTML juga masih dibagi menjadi 2 tag lagi. Tag pertama adalah `<head>` dan fungsinya sebagai tempat untuk memperkenalkan halaman HTML tersebut. Jadi, sekarang dokumen HTML Anda menjadi seperti berikut.

```
<html>
<head>
Ini informasi mengenai halaman web
</head>
</html>
```

Setelah head, bagian kedua adalah `<body>`. Di bagian inilah, Anda akan meletakkan teks dan berbagai hiasan serta file multimedia di halaman web. Tag `<body>` ini diletakkan di bawah penutup tag `<head>`.

```
<html>
<head>
Ini head
</head>
<body>
Semua teks body diisikan di sini
</body>
</html>
```

Mudah bukan, struktur halaman HTML memang sangat simple. Bahkan Anda yang baru belajar sekalipun akan langsung mudah mempraktikkannya.

### 1.4.2 Menambahkan Title

Title adalah judul halaman web. Judul ini berbeda dengan nama file HTML yang nanti dihasilkan. Karena nama file akan Anda buat saat menyimpan dokumen menggunakan editor teks, sementara title akan diisikan di dalam tag di kode HTML-nya.

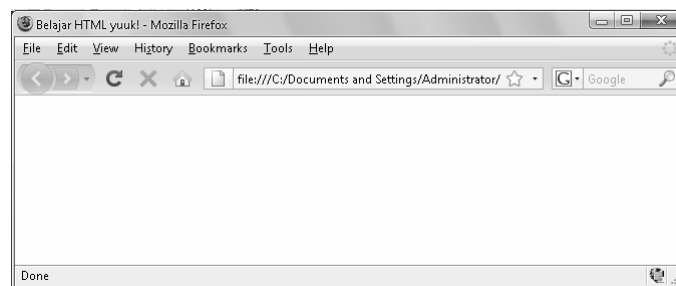
Untuk mendefinisikan title di dokumen HTML, tag yang dipakai adalah tag `<title>`. Misalnya untuk membuat judul “Belajar HTML yuuk!” maka Anda tinggal membuat kode berikut.

```
<title> Belajar HTML yuuk!</title>
```

Nah, tag `<title>` ini harus diletakkan di dalam tag `<head>`. Sehingga misalnya Anda akan membuat halaman kosong namun dengan judul di atas, maka tampilannya terlihat seperti berikut.

```
<html>
<head>
<title> Belajar HTML yuuk! </title>
</head>
<body>
</body>
</html>
```

Karena di tag `<body>` masih kosong, maka halaman ini tidak akan terlihat di browser. Yang terlihat hanya teks di tag saja. Berikut ini tampilannya jika Anda sudah menyimpan dokumen ke dalam format HTML.

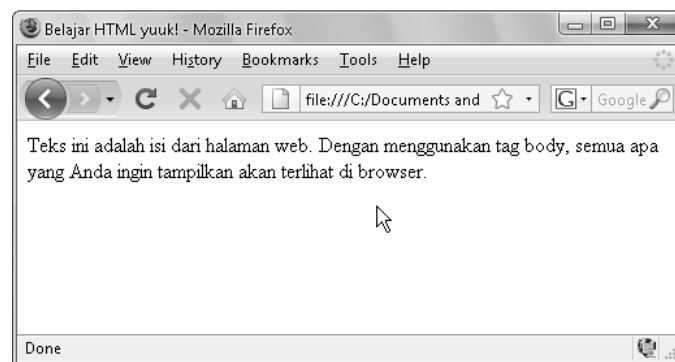


**Gambar 1.21** Hasil pembuatan halaman yang masih kosong namun dengan title yang sudah ada (lihat title bar di kiri atas)

### 1.4.3 Menambahkan Teks ke Body

Setelah title dari webpage dibuat, barulah Anda dapat memasukkan konten dari halaman web. Konten ini diletakkan di antara tag <body> dan </body>.

```
<html>
<head>
<title> Belajar HTML yuuk! </title>
</head>
<body>
Teks ini adalah isi dari halaman web.
Dengan menggunakan tag body, semua apa yang Anda ingin tampilkan
akan terlihat di browser.
</body>
</html>
```

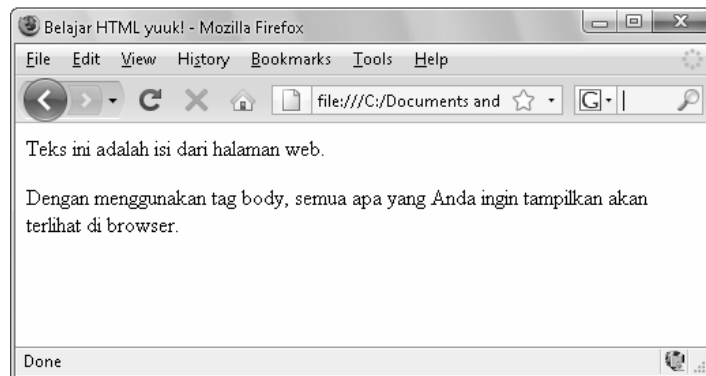


*Gambar 1.22 Isi tag body*

Dari gambar di atas, terlihat bahwa tampilan teks terus disambung walaupun di kodenya Anda sudah memisahkan teks menggunakan Enter (baris baru). Untuk membuat paragraf, Anda bisa menggunakan tag <p>.

```
<html>
<head>
<title> Belajar HTML yuuk! </title>
</head>
<body>
<p>Teks ini adalah isi dari halaman web. </p>
<p>Dengan menggunakan tag body, semua apa yang Anda ingin
tampilkan akan terlihat di browser. </p>
</body>
</html>
```





*Gambar 1.23 Konten body menjadi 2 paragraf karena tag <p>*

#### 1.4.4 Menggunakan Heading

Heading adalah bagian dari halaman yang memungkinkan dipakai untuk outline. Heading yang digunakan antara heading 1 sampai heading 6. Heading dibuat menggunakan tag <h1> sampai <h6>. Berikut ini contoh penggunaan heading. Walaupun sebenarnya Anda bisa membuat layout yang baik tanpa memakai heading, namun heading ini nanti juga penting ditinjau dari SEO (search engine optimization). Contoh kode pembuatan halaman untuk heading adalah:

```
<html>
<head>
<title>Heading, outline untuk halaman web</TITLE>
</head>
<body>
<h1>1. Mengenal Komputer</h1>
Komputer adalah komponen yang sangat penting peranannya
<h2>1.1 Komponen Komputer</h2>
Ada banyak komponen komputer
<h3>1.1.1 Penyimpan Data</h3>
Ada banyak komponen bisa dikategorikan sebagai storage device.
<h4>1.1.1.1 Hard Drive</h4>
Hard Drive adalah komponen utama penyimpan data yang memakai
platter
</body>
</html>
```



*Gambar 1.24 Pembuatan outline dengan heading*

#### 1.4.5 Membuat Link

Sebagai seorang netter, tentu Anda tahu bahwa halaman web umumnya selalu memiliki link. Link ini berguna untuk menuju ke tempat lainnya di belantara WWW. Apa sebenarnya kode untuk membuat link itu?

Kode untuk membuat link ternyata amat mudah. Anda tinggal menggunakan kode `<a>` untuk membuat link. Tapi penggunaan tag ini lebih kompleks daripada pembuatan tag lainnya karena Anda harus menentukan url yang akan dituju. Berikut ini format pembuatan link:

```
<a href="alamat url">Teks tertentu</a>
```

URL di sini digunakan misalnya:

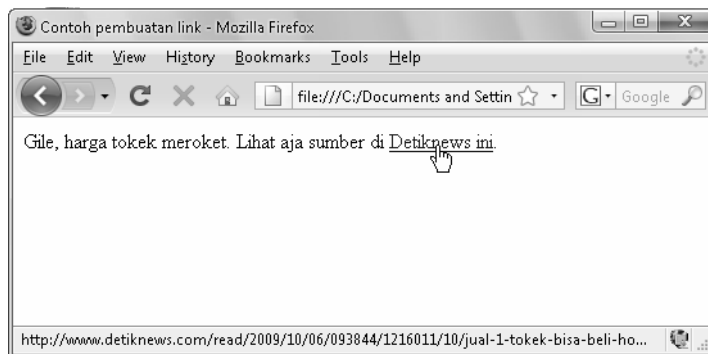
<http://www.yahoo.com>

atau

<http://www.detiknews.com/read/2011/10/06/093844/1216011/10/jual-1-tokek-bisa-beli-honda-jazz-nabung-ratusan-juta/>

Berikut ini contoh kode simpel untuk memperkenalkan Anda untuk membuat link:

```
<html>
<head>
<title>Contoh pembuatan link</title>
</head>
<body>
Gile, harga tokek meroket. Lihat aja sumber di <a
HREF="http://www.detiknews.com/read/2011/10/06/093844/1216011/10
/jual-1-tokek-bisa-beli-honda-jazz-nabung-ratusan-juta">
Detiknews ini</a>.
</body>
</html>
```



*Gambar 1.25 Contoh pembuatan link*

#### 1.4.6 Menambahkan Gambar

Mayoritas halaman web selalu memiliki gambar di dalamnya. Dengan gambar, halaman bisa terlihat berwarna dan tampilannya lebih bagus. Jika tidak ada gambar, halaman bisa terlihat membosankan. Untuk menambahkan gambar, kode HTML pun mendukungnya.

Pemasukan gambar diakomodasi oleh tag <img>. Berikut ini format standar untuk tag <IMG> dalam memasukkan gambar.

```

```

Misalnya Anda ingin memasukkan gambar dengan nama file “gambar.jpg” dengan lokasi ada di satu folder dari lokasi tempat file .html Anda berada. Maka Anda tinggal memasukkan kode:

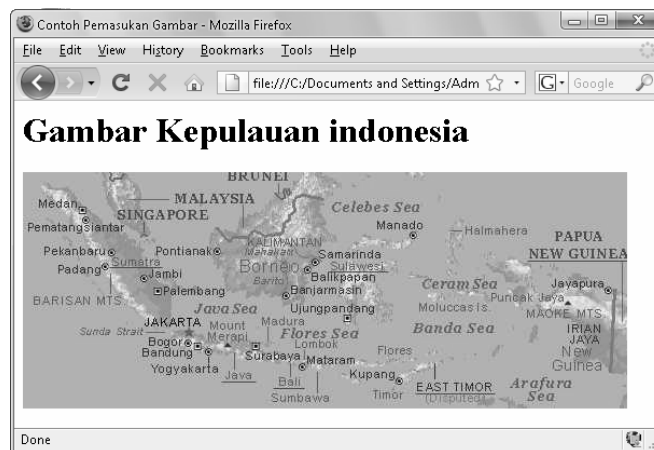
```

```

Contohnya kode berikut:

```
<html>
<head>
<title>Contoh Pemasukan Gambar</title>
</head>
<body>
<h1>Gambar Kepulauan Indonesia</h1>

</body>
</html>
```



*Gambar 1.26 Memasukkan gambar*

#### 1.4.7 Menambahkan Komentar

Komentar di HTML cukup mudah dibuat, yaitu tinggal memberikan pembuka `<!--` dan penutup `-->`. Sangat simpel bukan. Teks-teks yang ada di dalam komentar ini tidak akan ditampilkan di halaman browser.

## 1.5 Memformat Konten di HTML

Untuk memformat konten, ada beberapa tag yang perlu Anda pahami. Tag-tag ini terus berkembang seiring dengan perkembangan teknologi browser yang memungkinkan berbagai format baru diperkenalkan untuk mendapatkan hasil tampilan dan efek yang lebih banyak.

### 1.5.1 Mengatur Style Teks

Pertama kita akan belajar mengenai bagaimana membuat style dari text yang memungkinkan Anda membuat efek tebal, garis miring, garis bawah, monospace, dan coret. Efek-efek ini hampir semuanya diakomodasi oleh semua browser modern.

1. Untuk membuat efek tebal, digunakan tag `<b>` yang merupakan singkatan dari *bold*.
2. Untuk membuat efek garis miring, digunakan tag `<i>` yang merupakan singkatan dari *italic*.
3. Untuk membuat garis bawah, digunakan tag `<u>` yang merupakan singkatan dari *underline*.
4. Untuk membuat efek monospace, digunakan tag `<tt>` yang merupakan singkatan dari *typewriter text*.
5. Untuk membuat efek coret, digunakan tag `<strike>`, artinya coret.

Sebenarnya ada beberapa efek style lain dari pemrograman HTML, namun apakah style tersebut dikenali atau tidak, akan tergantung pada browser-nya. Namun browser modern umumnya mendukung 5 efek dasar di atas.

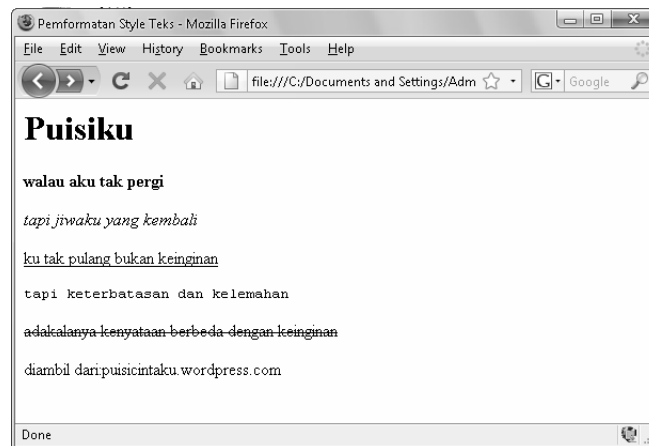
```
<html>
<head>
<title>Pemformatan Style Teks</title>
</head>
<body>
<h1> Puisiku</h1>
<p>
<b>walau aku tak pergi </b>
</p>
```

```

<p>
<i>tapi jiwaku yang kembali</i>
</p>
<p>
<u>ku tak pulang bukan keinginan</u>
</p>
<p>
<tt>tapi keterbatasan dan kelemahan</tt>
</p>
<p>
<strike>adakalanya kenyataan berbeda dengan keinginan</strike>
</p>
diambil dari:puisicintaku.wordpress.com
</body>
</html>

```

Apabila dilihat di browser, tampilan seperti berikut ini.



**Gambar 1.27 Pemformatan dengan mengganti style dari teks**

Untuk mengombinasikan format-format teks di atas, Anda tinggal meletakkan tag-tag tersebut pada teks yang diinginkan. Misalnya untuk membuat format garis miring, tebal dan coret secara bersamaan, Anda dapat menggunakan format berikut:

```

<b><i><strike>Tulisan ini mengalami multi format
</strike></i></b>

```

Berikut ini gambar hasil penulisan kode di atas.

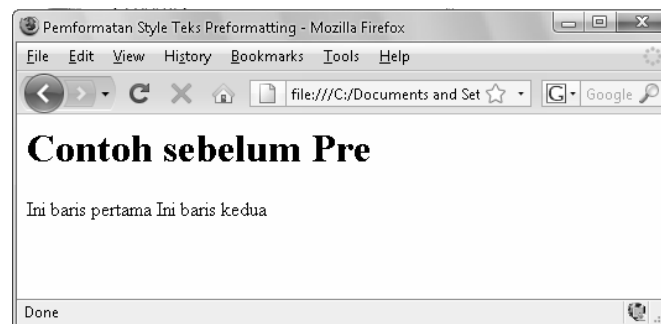
***Tulisan ini mengalami multi format***

**Gambar 1.28 Hasil penggunaan multiformat**

Selain 5 format dasar di atas, ada juga beberapa format tambahan yang masing-masing fungsinya berbeda. Yang pertama adalah <pre> di mana fungsinya untuk menggarap pemformatan seperti spasi, tab, dan enter yang sebelumnya tidak dikomodasi oleh browser.

Berikut ini contoh ilustrasi penggunaan tag <pre> yang merupakan singkatan dari preformatting.

```
<html>
<head>
<title>Pemformatan Style Teks Preformatting</title>
</head>
<body>
<h1>Contoh sebelum Pre</h1>
Ini baris pertama
Ini baris kedua
</body>
</html>
```



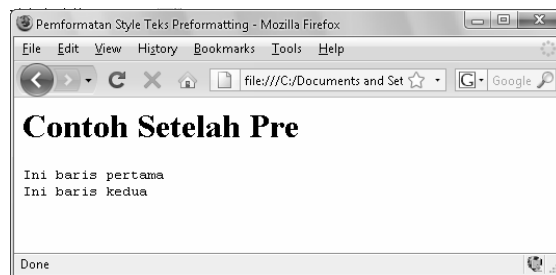
**Gambar 1.29 Sebelum digunakan fungsi <pre>**

Pada gambar di atas, walaupun di source code, tulisan “Ini baris pertama” berbeda baris dengan “Ini baris kedua”. Namun di browser, terlihat kedua baris tersebut digabung. Itu karena tidak ada pemformatan <pre>.

Jika diberi `<pre>`, maka tampilan akan seperti tampilan aslinya di kode sumbernya. Jadi, jika ada tab atau enter, akan diterapkan. Berikut ini kode jika di bagian body terlihat diberi kode `<pre>`.

```
<html>
<head>
<title>Pemformatan Style Teks Preformatting</title>
</head>
<body>
<h1>Contoh sebelum Pre</h1>
<pre>
Ini baris pertama
Ini baris kedua
</pre>
</body>
</html>
```

Setelah menggunakan `pre`, tampilan pemformatan akan menjadi sesuai dengan di source code-nya. Hanya saja, terlihat bentuk tulisan menjadi seperti tulisan monospace.



*Gambar 1.30 Tampilan halaman setelah di-pre*

### 1.5.2 Berpindah Baris

Untuk membuat tampilan pindah baris, Anda bisa menggunakan `break`. Tag `break` ini membuat hasil style teks masih bagus, tidak seperti `<pre>` yang membuat tulisan seperti monospace.

Tag untuk memindahkan teks ke baris berikutnya adalah tag `<br/>`. Tulisan `br` adalah singkatan dari `break` atau yang dimaksud adalah line break. Apa perbedaan `break` dengan tag `<p>` untuk pembuatan paragraf?



Jawabnya adalah paragraf akan selalu memiliki jarak dengan paragraf di bawahnya. Adapun pemindahan baris dengan break tidak memberikan jarak dengan bawahnya. Jadi dengan break, tulisan teks masih akan dianggap satu paragraf.

Berikut ini contoh sebelum adanya break dan hanya menggunakan paragraf.

```
<html>
<head>
<title>Sebelum digunakan Break</title>
</head>
<body>
<h1>Sebelum Break</h1>
<p>
Suatu waktu di suatu masa
tersebutlah kisah seorang dara
</p>
<p>
Dara elok berparas rembulan
disayang dimanja oleh Sang Raja
</p>
<p>
Dara cantik sudah remaja
tibalah waktu mencari pendamping
</p>
</body>
</html>
```



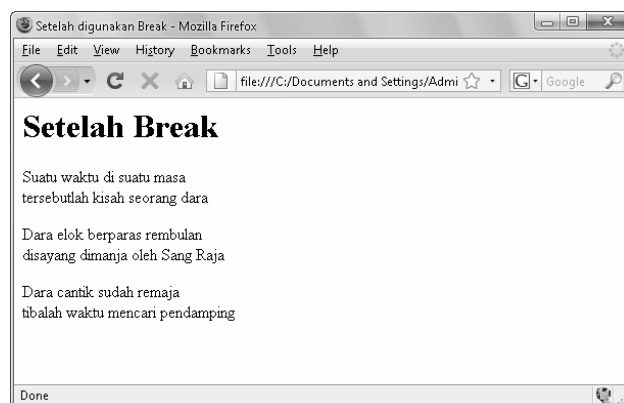
**Gambar 1.31** Sebelum diterapkan break

Adapun berikut ini merupakan kode setelah di dalam paragraf diberi tag `<br/>` untuk memisahkan baris. Dari tampilan baru ini, Anda juga akan dapat melihat perbedaan antara tag `<p>` dan tag `<br>`.

```

<html>
<head>
<title>Setelah digunakan Break</title>
</head>
<body>
<h1>Setelah Break</h1>
<p>
Suatu waktu di suatu masa
<br/>
tersebutlah kisah seorang dara
</p>
<p>
Dara elok berparas rembulan
<br/>
disayang dimanja oleh Sang Raja
</p>
<p>
Dara cantik sudah remaja
<br/>
tibalah waktu mencari pendamping
</p>
</body>
</html>

```



*Gambar 1.32 Setelah diterapkan break*

### 1.5.3 Memasukkan Garis Horizontal

Garis horizontal adalah garis yang membujur secara horizontal di halaman web. Ini dapat dimasukkan dengan menggunakan tag `<hr/>` di kode Anda. HR ini adalah singkatan dari *horizontal ruler*. Garis horizontal ini adalah salah satu teknik yang bisa dipakai untuk memisahkan atau mempartisi konten halaman web Anda.

Langsung saja, berikut ini contoh kode untuk memasukkan garis horizontal:

```
<html>
<head>
<title>Horizontal Line</title>
</head>
<body>
<h1>Puisi dengan HR</h1>
<hr/>
hidup ini sesungguhnya mudah dan sederhana jika anda tidak
banyak tuntutan.
karena sesungguhnya hidup ini penuh dengan kebahagiaan
jika anda tak dikuasai oleh ilusi, ambisi dan keakuan
<hr/>
</body>
</html>
```

Jika dibuka di browser, tampilannya terlihat seperti berikut, ada bagian horizontal ruler di atas dan bawah teks.



**Gambar 1.33** Contoh penggunaan Horizontal Line

#### 1.5.4 Membuat Aneka Macam List

Sering kali di sebuah website, diperlukan pembuatan daftar atau list. Daftar ini ada berbagai macam variasinya di pemrograman HTML. Yaitu *ordered list*, *unordered list* dan *definition list*.

Untuk membuat daftar berurutan Anda perlu 2 kode:

- Semua list dibuka dengan tag `<ol>` dan ditutup dengan tag `</ol>`. Tag OL ini adalah singkatan dari ordered list.

- Tiap item dari list yang ada dibuka dengan tag `<li>` dan ditutup dengan tag `</li>`. LI ini adalah singkatan dari *list item*.

Setelah list bernomor sebelumnya, Anda dapat juga bisa membuat jenis yang kedua, yaitu list bullet. Yaitu berjenis unordered list. Nama bullet ini disebabkan browser menampilkan ikon bulat-bulat di sebelah kiri tiap list item yang dimasukkan. Kodenya hampir mirip dengan ordered list, di mana:

- Semua list dibuka dengan tag `<ul>` dan ditutup dengan tag `</ul>`. Tag UL ini adalah singkatan dari *unordered list*.
- Tiap item dari list dibuka dengan tag `<li>` dan ditutup dengan tag `</li>`.

Berikut ini kode yang mengandung tipe ordered list dan unordered list:

```
<head>
<title>Unordered dan Orderd List</title>
</head>
<body>
<h1>Daftar Komponen PC</h1>
<ul>
<li>Case</li>
<li>Mother Board </li>
<li>CPU (Central Processing Unit) </li>
<li>RAM (Random Access Memory)</li>
<li>Hard Disk Drive</li>
<li>Video Display Adapter</li>
<li>Monitor </li>
<li>Input Device</li>
<li>Printer</li>
</ul>
<h1> Cara merakit PC</h1>
<ol>
<li>Pasang Prosesor ke Motherboard</li>
<li>Pasang Memory ke Motherboard</li>
<li>Pasang Motherboard ke Casing</li>
<li>Pasang Powersupply dan Kabelnya</li>
<li>Pasang kabel controller depan</li>
<li>Pasang hard drive dan drive optik</li>
<li>Oke, selesai</li>
</ol>
</body>
</html>
```

Jika dilihat di browser, tampilan list ini terlihat seperti berikut.



**Gambar 1.34** Hasil tampilan kode unordered dan ordered list di browser

Tipe list ketiga adalah definition list. Tipe ini fungsinya memformat teks dalam bentuk seperti kamus, sehingga Anda dapat menuliskan kata dan mendefinisikan artinya. Jadi, tipe ini memiliki 2 bagian, yaitu istilah dan arti dari istilah tersebut. Namun tag-nya ada tiga, yaitu:

- Kata yang akan dijelaskan menggunakan tag `<dt>` yang merupakan singkatan dari *definition term*.
- Kata penjelasannya menggunakan tag `<dd>` yang artinya *definition description*.
- Semua tag ini diawali oleh tag `<dl>` yang merupakan singkatan dari *definition list*.

Contoh kode HTML yang menampilkan definition list ini adalah:

```
<head>
<title>Definition List</title>
</head>
<body>
<h1> Definisi Komponen PC</h1>
<dl>
<dt>Case </dt>
    <dd>Tempat untuk menyimpan semua komponen komputer</dd>
<dt>Mother Board</dt>
```

```

        <dd>Tempat untuk menancapkan semua komponen dan
        perkabelan</dd>
<dt>CPU (Central Processing Unit)</dt>
        <dd> Otaknya komputer</dd>
<dt>RAM (Random Access Memory) </dt>
        <dd>Tempat penyimpanan data sementara</dd>
<dt>Hard Disk Drive</dt>
        <dd>Tempat penyimpanan data permanen</dd>
<dt>Video Display Adapter </dt>
        <dd>Tempat me-render gambar</dd>
<dt>Monitor</dt>
        <dd>Tempat menampilkan gambar</dd>
<dt>Input Device</dt>
        <dd>Tempat memasukkan masukan ke komputer</dd>
</dl>
</body>
</html>

```

Jika dilihat di browser, terlihat webpage seperti ini.



**Gambar 1.35** Webpage yang berisi kode untuk definition list

List-list di atas dapat juga dikombinasikan. Caranya sama seperti ketika meng-combo-kan style pada teks. Jadi, Anda hanya perlu memainkan peletakan tag di dalam tag list lainnya.

```

<dl>
<dt>Komponen-komponen komputer yang penting:</dt>
<dd>
List-nya adalah seperti berikut:
<ul>
<li>Monitor</li>
<li>Sistem operasi</li>

```

```
<li>CPU</li>
</ul>
</dd>
</dl>
```

Jika dilihat di browser, tampilan unordered list akan ada di dalam bagian definition description. Karena tag <ul> ada di dalam tag <dd>.

### 1.5.5 Membuat Link Lanjutan

Di subbab sebelumnya, Anda sudah belajar membuat link. Sebenarnya pemformatan link ada beberapa variasi. Pertama adalah tipe link internal. Ini merupakan tipe link di mana halaman yang mengandung link dan halaman yang dituju terletak pada satu domain dan letaknya bisa direlatifkan.

Misalnya Anda punya 2 halaman:

<http://www.domain-anda.co.cc/index.html>

dan Anda ingin membuat link dari halaman index.html di atas untuk menuju ke:

<http://www.domain-anda.co.cc/bukutamu.html>

Dari URL-nya, kedua halaman tersebut terletak di satu folder. Ini memungkinkan Anda membuat email dengan menentukan alamat relatif dari halaman kedua melalui halaman pertama. Halaman kedua `bukutamu.html` tersebut dapat dispesifikasikan lokasinya dengan menggunakan nama file-nya saja, karena lokasinya satu folder. Jadi, Anda bisa menggunakan kode berikut:

```
Ayo isi buku tamu di <a href="bukutamu.html">Buku tamu gw</A>
```

Jenis lain adalah link internal. Untuk membuat link internal, Anda harus membuat dahulu anchor di dalam website Anda. Tempat yang di-anchor ini nanti dapat diacu dengan link. Jadi, anchor fungsinya adalah “menandai tempat” sehingga bisa diacu oleh link.

Misalnya Anda ingin memberi nama di bagian tertentu di webpage dengan nama “iklan”. Maka Anda membuat nama anchor seperti berikut:

```
<a name="iklan">Kotak iklan</a>
```

Maka link ke kotak iklan tersebut dapat dibuat dari halaman yang sama (misalnya dalam bentuk daftar isi halaman tersebut), dengan menggunakan kode HTML seperti berikut:

```
<a href="#iklan">Kotak Iklan</a>
```

Jika link itu dibuat dari halaman lain, maka Anda bisa menggunakan kode HTML seperti berikut:

```
<a href="http://www.domain-  
anda.co.cc/file_tertentu.htm#iklan">Lihat kotak iklannya di  
sini</a>
```

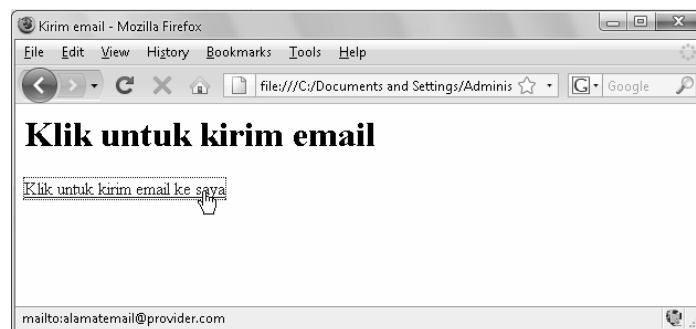
### 1.5.6 Membuat Link Email

Sering kali Anda perlu untuk membuat link yang memungkinkan user dapat langsung mengklik untuk mengirimkan email ke account email Anda. Ini bisa juga langsung diakomodasi oleh HTML. Namun hati-hati, teknik ini juga bisa mendorong robot dari spammer untuk menuju ke website Anda. Karena itu jangan gunakan kecuali jika terpaksa.

Untuk membuat link langsung ke email, format kode yang dapat Anda gunakan seperti berikut:

```
<a href="mailto:alamatemail@provider.com">Klik untuk kirim email  
ke saya</A>
```

Implementasinya tinggal Anda mengklik link tersebut, maka terbuka software mail client Anda, seperti Outlook atau Thunderbird dan kemudian Anda bisa mengirim email.



*Gambar 1.36 Link yang memungkinkan pengiriman email*



## 1.6 Menggunakan Gambar di Halaman

Anda sudah mengetahui bahwa memasukkan gambar dapat diakomodasi dengan kode berikut:

```

```

Kode src di atas adalah singkatan dari source. Sekarang kita akan belajar beberapa utak-atik sederhana tentang gambar di halaman web.

### 1.6.1 Mengatur Alignment Teks dan Gambar

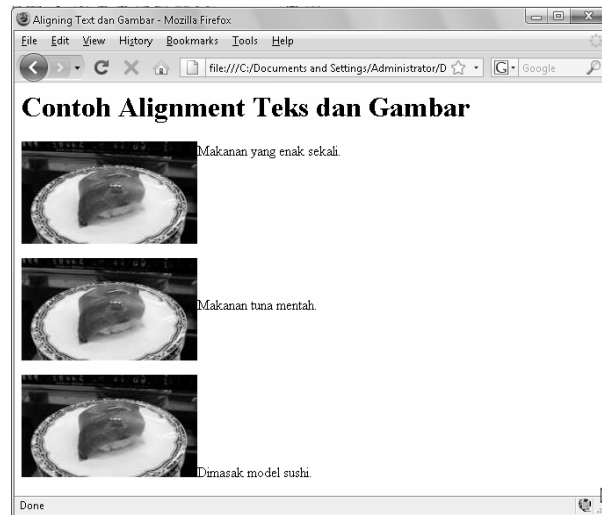
Dalam prakteknya, gambar sering kali disandingkan dengan teks. Untuk itu Anda perlu mengatur bagaimana alignment teks terhadap gambar. Caranya dengan menambahkan tag <align> di dalam tag <img>. Format kodenya seperti berikut:

```

```

Maksud tulisan top|middle|bottom adalah Anda bisa menentukan apakah alignment akan top (atas), middle (tengah) atau bottom (bawah). Langsung saja, berikut ini contoh alignment teks beserta tampilannya jika dilihat di browser.

```
<html>
<head>
<title>Aligning Text dan Gambar</title>
</head>
<body>
  <h1> Contoh Alignment Teks dan Gambar </h1>
  <p>
    Makanan yang enak sekali.
  </p>
  <p>
    Makanan tuna
    mentah.</p>
  <p>
    Dimasak model
    sushi.</p>
</body>
</html>
```



*Gambar 1.37 Alignment teks dan gambar, dalam bentuk top, middle dan bottom*

### 1.6.2 Memberikan Teks Alternatif

Di awal-awal kemunculan bahasa HTML, tidak semua browser mampu menampilkan gambar. Karena itu bahasa HTML mengakomodasi teks alternatif yang memungkinkan browser yang tidak dapat menampilkan gambar tersebut untuk menampilkan teks.

Sekarang walaupun hampir semua browser mampu menampilkan gambar, namun teks alternatif ini juga masih tetap berguna. Karena diperlukan dalam kampanye SEO (search engine optimization). Alt ini biasanya diisi dengan kata-kata kunci berkaitan dengan gambar yang ditampilkan.

Formatnya seperti berikut:

```

```

Jika browser dibuka, maka tulisan alternatif ini terlihat di kotak gambar jika seandainya gambar gagal dibuka. Dan jika pointer mouse sedang berada di atas (hover) gambar.

### 1.6.3 Membuat Link Gambar

Gambar memang multifungsi. Tidak cuma sebagai media untuk mempercantik dan menghias halaman serta menambah ilustrasi di halaman. Namun juga bisa dipakai untuk membuat link. Link yang berupa gambar akan lebih menarik pengunjung. Terutama jika gambarnya menarik dan senada dengan informasi yang akan ditampilkan pada gambar.

Tekniknya sebenarnya simpel. Hanya merupakan gabungan dari tag `<a>` dan di tengahnya ada tag `<img>` untuk menampilkan gambar. Jadi, gambarnya menjadi link. Formatnya seperti berikut:

```
<a href="url_dari_link"></a>
```

Contohnya kode berikut:

```
<html>
<head>
<title>link dengan gambar</title>
</head>
<body>
  <h1> klik pada gambar untuk menuju halaman lain </h1>
  <a href="http://detikfood.com">
</a>
</body>
</html>
```

Berikut ini tampilannya saat dilihat di browser.



*Gambar 1.38 Link dengan gambar*

## 1.7 Membuat Tabel

Tabel adalah komponen yang cukup penting di halaman web. Selain untuk membuat tabel untuk menampilkan data, tabel di sini juga penting untuk me-layout halaman. Misalnya saat menampilkan form untuk menerima input dari pengguna.

Jadi, fungsinya yang secara langsung memang tidak terlalu penting (seperti membuat tabel dalam arti eksplisit), tapi karena tabel mampu digunakan untuk membuat tampilan form menjadi teratur, maka tabel ini akan sering sekali ditemui dalam membuat halaman web.

Untuk membuat tabel, tag pembungkusnya adalah tag `<table>`. Di tabel, ada 2 istilah yang perlu diperhatikan, yaitu row (artinya baris) yang dibuat dengan tag `<tr>`. Dan tiap baris ini akan terdiri atas sel-sel data/*table data* yang dibuat dengan tag `<td>`.

Dan sel-sel di tabel ini kemudian dapat diisi dengan berbagai macam data, seperti teks, gambar, list, paragraf, form, ruler, bahkan bisa dimasuki lagi tabel di dalamnya. Berikut ini contoh kode untuk membuat tabel di HTML.

```
<html>
<head>
<title>tabel di html</title>
</head>
<body>
<h1> tabel, bagian penting dari kode html </h1>
<table border="1">
<tr>
<td>baris 1, kolom 1</td>
<td>baris 1, kolom 2</td>
<td>baris 1, kolom 3</td>
</tr>
<tr>
<td>baris 2, kolom 1</td>
<td>baris 2, kolom 2</td>
<td>baris 2, kolom 3</td>
</tr>
<tr>
<td>baris 3, kolom 1</td>
<td>baris 3, kolom 2</td>
<td>baris 3, kolom 3</td>
</tr>
</table>
</body>
</html>
```

Jika dilihat di browser, tampilannya seperti berikut ini.



*Gambar 1.39 Tabel 3 baris dan 3 kolom*

Saat membuat tabel di atas, Anda menggunakan atribut border. Atribut tersebut merupakan atribut yang digunakan untuk menentukan lebar batas dari tabel. Jadi, jika tabel hanya digunakan untuk membantu me-layout peletakan gambar, teks, atau form, maka border tidak digunakan dan tabel hanya dibuka dengan <table> saja.

Anda juga bisa memakai heading untuk tiap bagian atas dari tabel dengan menggunakan tag <th> di dalam baris atau tag <tr> yang pertama.

Format untuk membuat heading di tabel adalah seperti ini.

```
<tr>
<th>Heading pertama</th>
<th>Heading di kolom kedua</th>
<th>Heading di kolom ketiga, dst</th>
</tr>
```

\*\*\*